This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

Claims 1 – 30 (Canceled).

Claim 31 (New): A method for a virtual machine in which statically compiled code may be securely executed by a virtual machine by means of the compiler or code generator, the method comprising the steps of:

a) verifying that the intermediate or byte-code representation of the program is safe;

b) forming a secure hash describing the generated code;

c) forming a secure hash describing the byte code;

d) digitally signing the secure hashes of the generated code and the byte code;

the executing virtual machine reuses this code by

e) verifying that the secure hash of the byte-code matches the digitally signed secure hash for the byte-code;

f) verifying that the secure hash of the generated code matches the digitally signed secure hash for the generated code;

g) loading and executing the generated code.

Claim 32 (New): A method for linking separately statically compiled code at run-time within a virtual machine by modifying the code, the method comprising the steps of

using a compiler to perform the steps of

a) maintaining symbolic entries for externally referenced symbols; and

b) maintaining a mapping from locations in the generated code that reference external symbols to the symbolic entry for that symbol

and the virtual machine, before the code is executed, performs the steps of

c) using the mapping and symbolic entries created by the compiler to generate direct references in the generated code to the externally referenced symbols that have been resolved by the virtual machine; and

d) performing the default action on those external symbols that have not been resolved.

Claim 33 (New): A method for updating statically generated code (C), at run-time, when separately compiled code (S), which contained symbols referenced by C changes, the method comprising the steps of:

having the compiler generating the code for S to a) associate with method and data names, or signatures, in S a secure hash of the name of the method and data names or signatures in S with the compiler for C recording the secure hash for the byte code corresponding to any S that affects the code generated for C; and

having the virtual machine executing C to

c) check if any byte codes associated with any names in the S relied upon by C have changed by comparing the secure hash of the names associated with S with the secure hash stored for this byte code in C, and

d) dynamically recompile the byte codes associated with C if any byte codes associated with S have changed.

Claim 34 (New): A method for maintaining full compliance with a language requiring dynamic compilation without requiring the overhead of a just-in-time compiler to be present

in the virtual machine, while enabling the use of statically generated code (C) for some byte code that depends on some byte code (S) that may be separately compiled,

having the compiler generating code for S

a) associate with S a secure hash of the byte code associated with S;

having the compiler for C to

b) record the secure hash for the byte code corresponding to any S that affects the code generated for C; and

having the virtual machine executing C to

c) check if any byte codes associated with any code S relied upon by C have changed by comparing the secure hash of the byte code associated with S with the secure hash stored for this byte code with C; and

d) interpret the byte codes corresponding to C.